In Silico Design and Analysis of Biological Systems

The Calculus of Looping Sequences and other Formalisms

Angelo Troina

Dipartimento di Informatica, Università di Torino

29 January 2008

Towards Systems Biology

The Human Genome Project:

- produced a huge amount of data about the **structure** of living matter;
- would have been impossible without computers, algorithms and syntax to model the structures.

Less is known about the biological **function** (behaviour) of cells and their components:

- the interest moved from **structure** to **functionality**
- growth of a new paradigm that moves from the classical **reductionist** approach to a system level understanding of life (**Systems Biology**, [Hood 2000]).

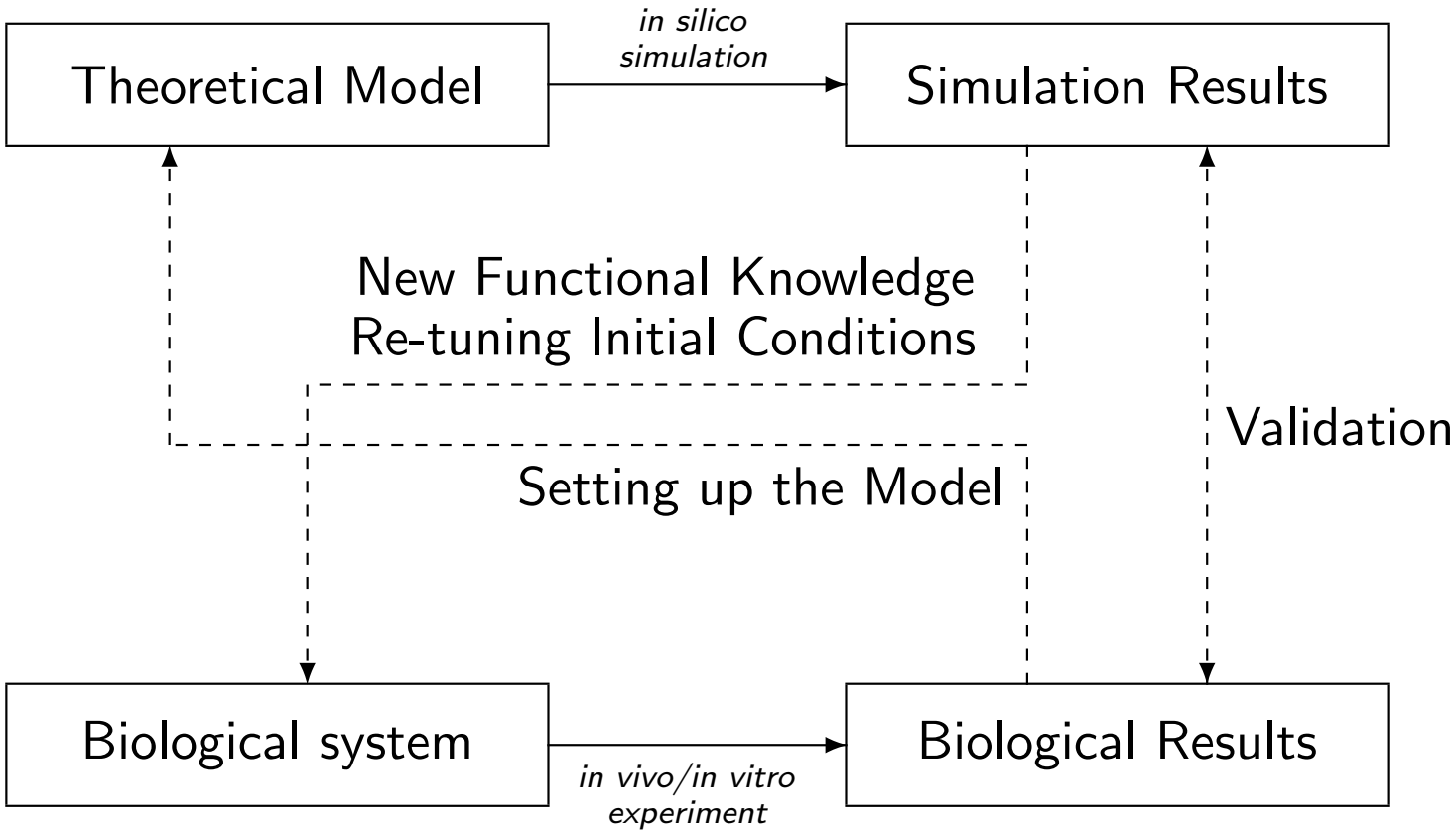
Since moving from **structure** to **functions** amounts at equipping a **syntax** with a **semantics**, Computer Science appears to be essential also for understanding the behaviour of living organisms.

Systems Biology

Systems Biology is a relatively new biological study field that focuses on the *systematic* study of complex interactions in biological systems, thus using a new perspective (**integration** instead of **reduction**) to study them.

- Used to obtain, integrate and analyse complex data from multiple experimental sources (Genomics, Proteomics, etc.) using interdisciplinary tools.
- Aims at the development and application of data-analytical and theoretical methods, mathematical modeling and computational simulation techniques to the study of biological systems.

From Biology to Computer Science (and back)



Computing Models Applied to Systems Biology

- Lambda-calculus [Fontana & Buss, 1996];
- Petri nets [Matsuno et al., 2000];
- Process Calculi:
  - Biological π-calculus [Regev, Shapiro et al., 2001/2002];
  - BioAmbients [Regev et al. 2004];
  - Brane Calculi [Cardelli, 2005];
  - Beta-binders [Priami & Quaglia, 2005];
  - BioPEPA [Hillston et al., 2006];
- Rewrite Systems:
  - P-Systems [Paun, 1998];
  - κ-calculus [Danos & Laneve, 2003];
  - CLS [Barbuti et al. 2005];
  - Stochastic Bigraphs [Krivine et al., 2007];
- Statecharts [Harel et al., 2003];
- Hybrid Automata [Mishra et al. 2006]; ...

Pi-Calculus Applied to Biology

A. Regev, W. Silverman, E. Shapiro. **Representation and Simulation of Biochemical Processes Using the pi-Calculus Process Algebra**. Pacific Symp. Biocomp., WSP 2001.

C. Priami, A. Regev, W. Silverman, E. Shapiro. **Application of a stochastic name-passing calculus to representation and simulation of molecular processes**. Information Processing Letters, 80:25-31, 2001.

Key features of the *pi*-calculus applied to biology:

- a molecule is a pi-calculus process;
- binding is communication;
- molecular recognition is name matching;

$$Na = \overline{a}\langle e \rangle.Na^{+} \qquad Na^{+} = a(x).Na$$
$$Cl = a(x).Cl^{-} \qquad Cl^{-} = \overline{a}\langle e \rangle.Cl$$

$$Na|Cl = \overline{a}\langle e \rangle.Na^{+}|a(x).Cl^{-} \rightarrow Na^{+}|Cl^{-}$$

Simulation Tools

There is a variety of semantically-based tools for the analysis of the behaviour of concurrent processes, e.g. for searching the state space or checking semantic equivalences and preorders:

- Concurrency Workbench(`www.dcs.ad.ac.uk/home/cwb`);
- Mobility Workbench (`www.it.uu.se/research/group/mobility/mwb`).

Tools that explicitly manage stochastic aspects:

- PEPA workbench (`www.dcs.ed.ac.uk/pepa/`);
- TwoTowers (`www.uniurb.it/bernardo/twotowers/`).

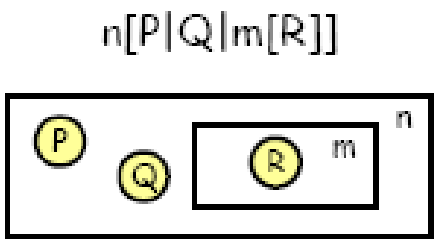
Simulators implementing Gillespie's algorithm (1977) over executions of terms of the stochastic pi-calculus:

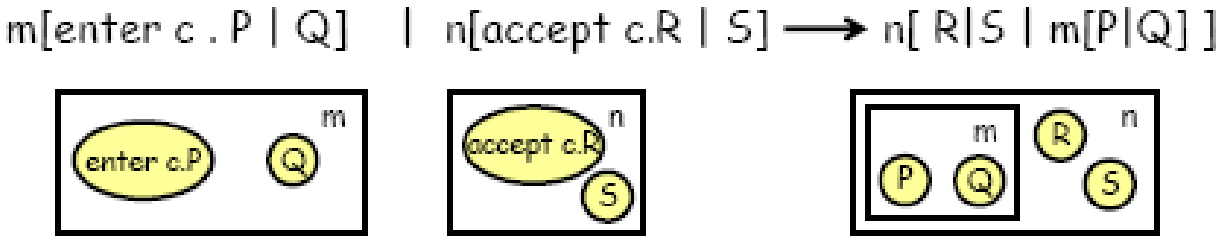
- BioSpi (`www.wisdom.weizmann.ac.il/~biopsi/`);
- SPiM (`www.doc.ic.ac.uk/~anp/spim/`).

BioAmbients (1)

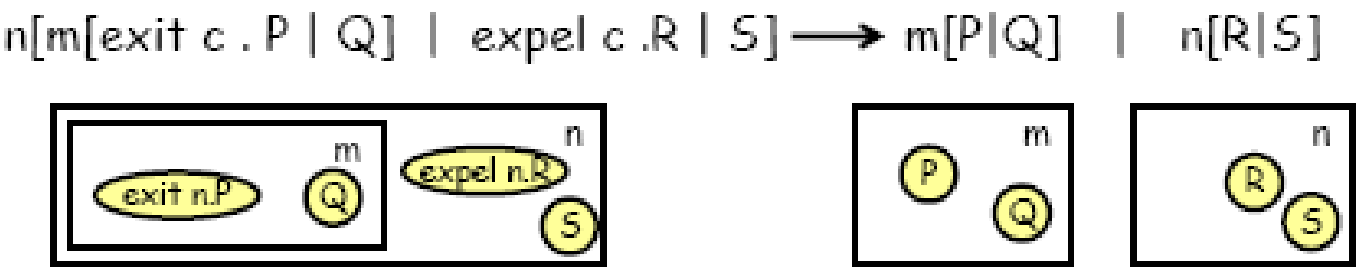
A. Regev, E. M. Panina, W. Silverman, L. Cardelli and E. Shapiro. **BioAmbients: An Abstraction for Biological Compartments**. Theor. Comp. Sci. 325(1):141-167, 2004.

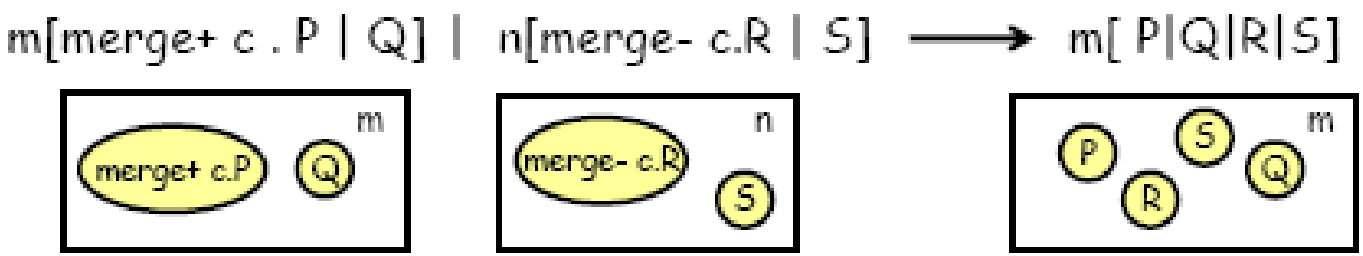
Pi-Calculus extended with capabilities for compartments.





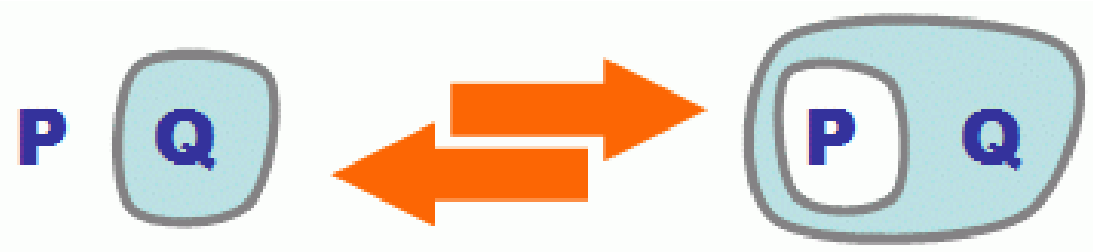
BioAmbients (2)





PEP - Brane Calculus (1)

L. Cardelli. **Brane Calculi. Interactions of Biological Membranes**. Proc. of CMSB'04, Springer LNCS 3082, 2005.

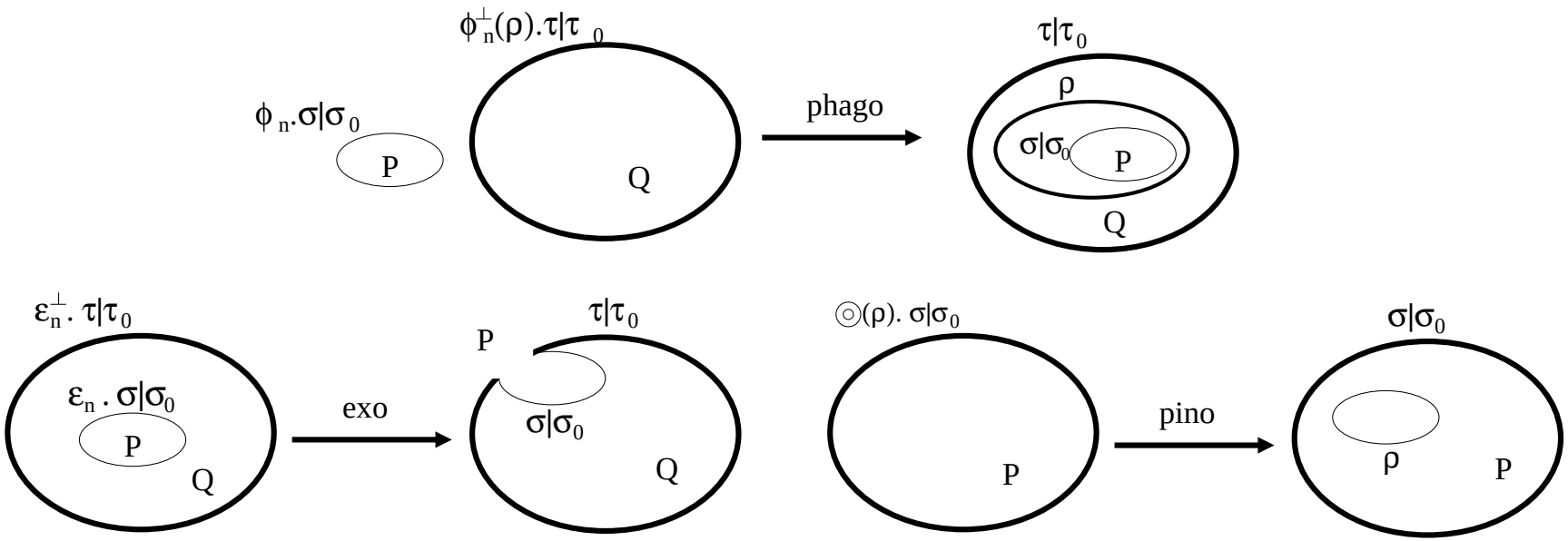


The phago/exo/pino (PEP) calculus: **Syntax**

$$
\begin{array}{rcl}
P, Q, R \ldots & ::= & \diamond \mid P \circ P \mid !P \mid \sigma(\!|P|\!) \quad \text{Systems} \\
\sigma, \tau, \rho, \ldots & ::= & \mathbf{0} \mid \sigma|\sigma \mid !\sigma \mid a.\sigma \quad \text{Branes} \\
a, b, c, \ldots & ::= & \phi_n \mid \phi_n^{\perp}(\sigma) \mid \varepsilon_n \mid \varepsilon_n^{\perp} \mid \circledcirc(\sigma) \quad \text{Actions}
\end{array}
$$

PEP - Brane Calculus (2)

The phago/exo/pino (PEP) calculus: **Semantics**



P-Systems

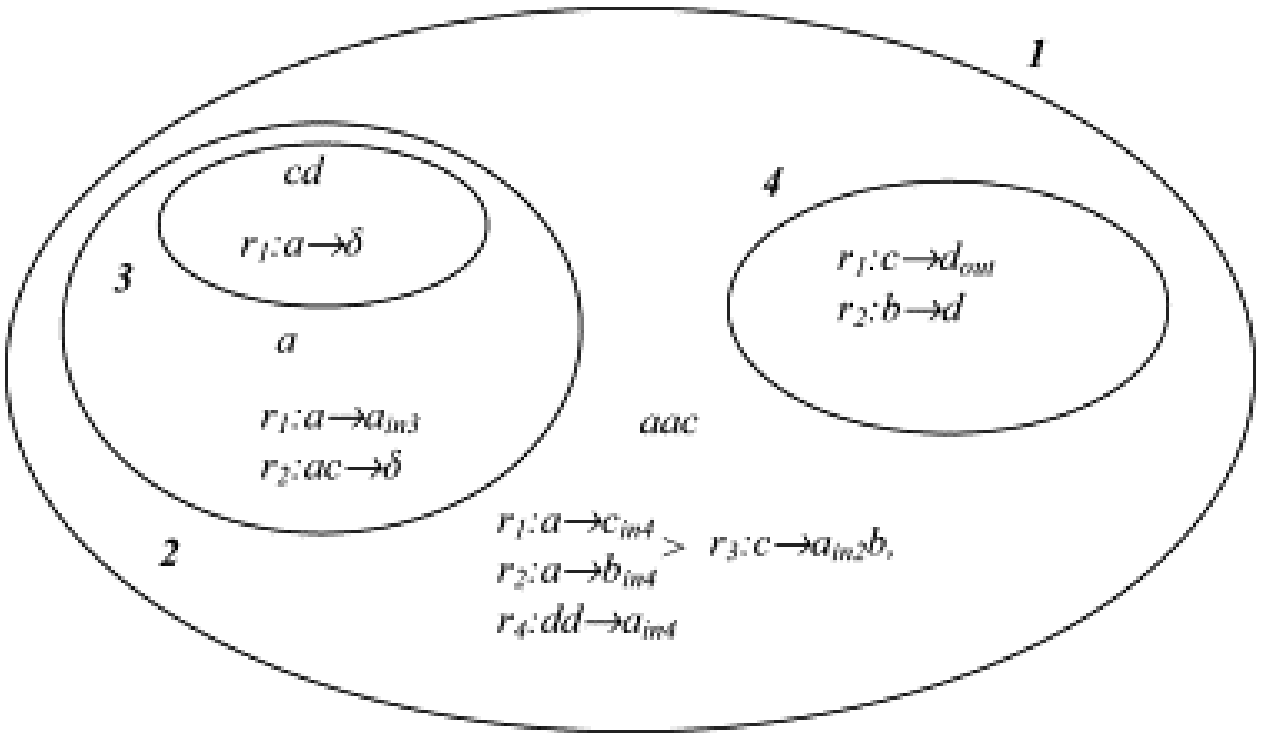
G. Paun. **Membrane Computing: An Introduction**. Natural Computing Series, Springer, 2002.

P-Systems' key features:

- components are delimited by membranes;
- each membrane contains a multiset of **objects** (DNA strands, molecules, etc.) and a set of **evolution rules** (chemical reactions);
- evolution rules operate on objects, by *modifying* them, and on the membrane structure, by *dissolving*, *creating* or *dividing* membranes;
- (nondeterministic) maximal parallel executions;
- molecules can move from a region to another by passing through the membranes (driven by evolution rules).

The grammatical complexity/expressivness of P-Systems has been investigated with respect to certain restrictions: how many membranes/objects/rules are needed to result in computational (Turing) universality?

P-Systems: An Example



Simulators for P-Systems

Metabolic Algorithm [Bianco, Fontana et al., 2006] gives a deterministic simulator for P-Systems: the goal is to simulate biological systems managing populations of objects instead of single elements.

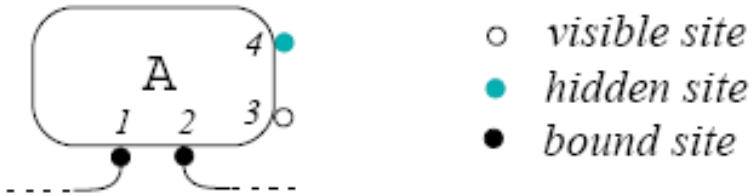
Dynamical Probabilistic P-System [Pescini, et al., 2006] a stochastic simulator for P-Systems where evolution rules are associated with a rate parameter and dynamics are driven by the law of mass action.

κ-Calculus

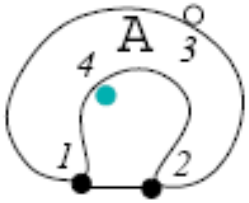
V. Danos and C. Laneve. **Formal Molecular Biology**. Theor. Comp. Sci. 325(1): 69-110, 2004.

A formal model (based on graph rewriting) to describe proteins complexation and decomplexation.

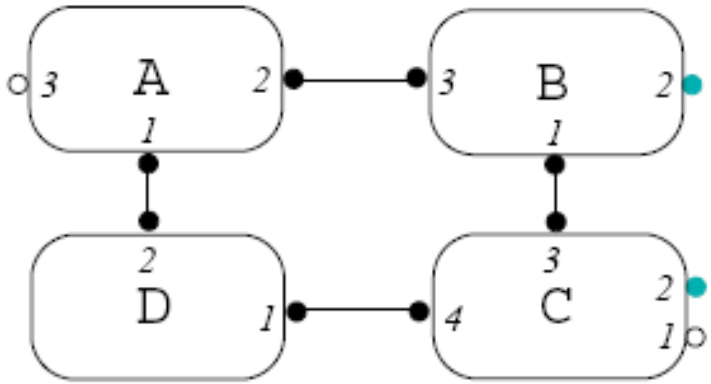
The calculus is equipped with a graphical notation, and is based on the concept of shared names to model bindings among proteins.



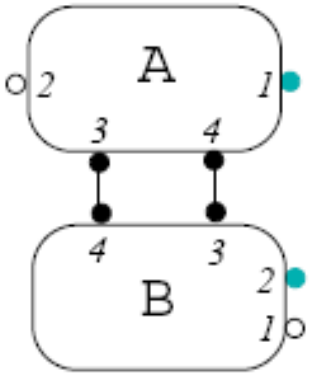
κ-Calculus: Complexes



a self−complexation



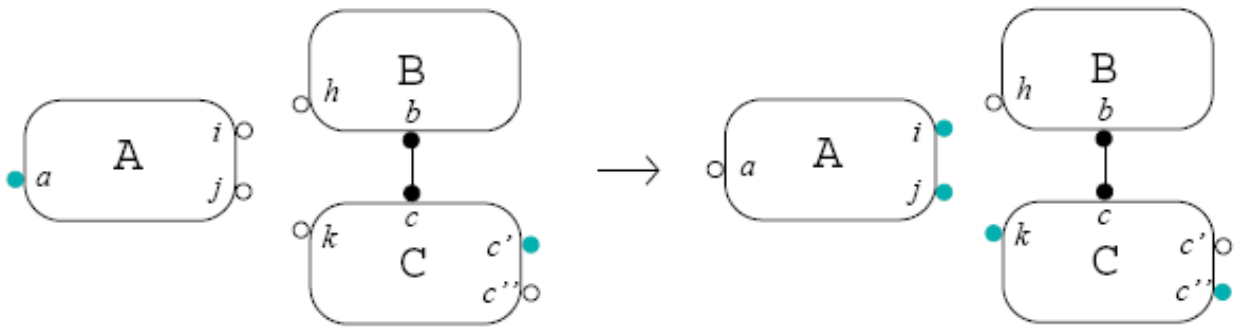
a ring−complex



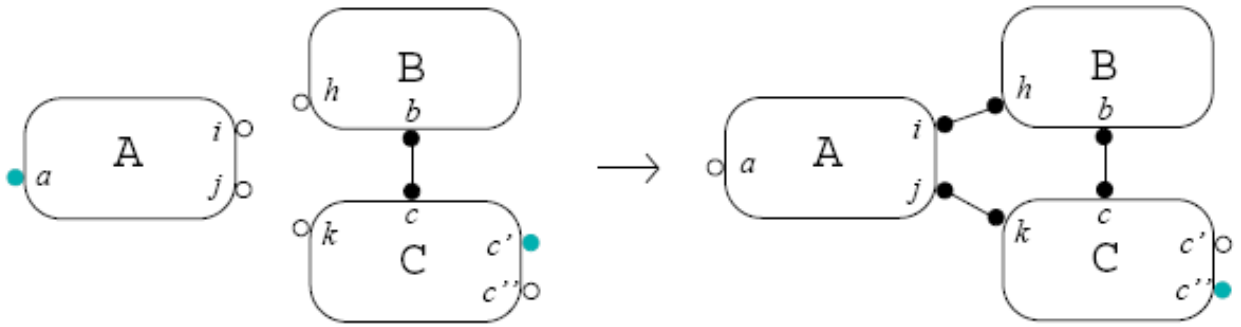
a double contact

κ-Calculus: Rewrite Rules

Two kinds of reactions deserve special attention: *activations*, when connections are left untouched and only states change, and *complexations* when the right hand side is connected.



activation



complexation

The Calculus of Looping Sequences (CLS)

We assume an alphabet $\mathcal{E}$. **Terms** T and **Sequences** S of CLS are given by the following grammar:

$$\begin{array}{rcl} T & ::= & S \mid (S)^L \rfloor T \mid T \mid T \\ S & ::= & \epsilon \mid a \mid S \cdot S \end{array}$$

where a is a generic element of $\mathcal{E}$, and ϵ is the empty sequence.

The operators are:

$S \cdot S$: Sequencing

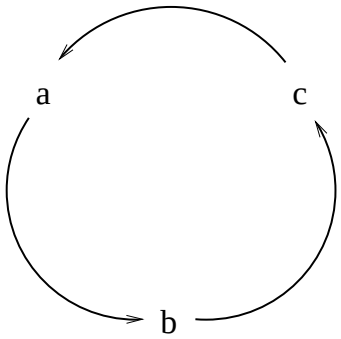
$(S)^L \rfloor T$: Looping and Containment (*binary* operator)

: S is closed, can rotate and contains T

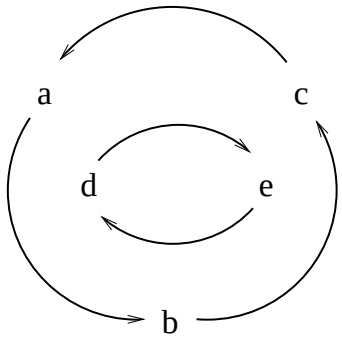
$T \mid T$: Parallel composition (juxtaposition)

Example of Terms

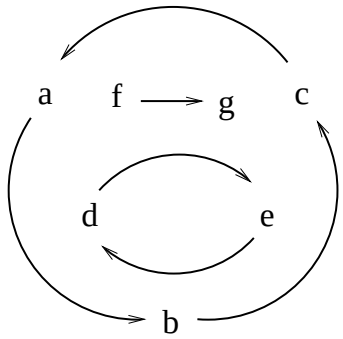
(i)



(ii)



(iii)



$$(i) \qquad \left(a \cdot b \cdot c\right)^{L} \rfloor \epsilon$$

$$(ii) \qquad \left(a \cdot b \cdot c\right)^{L} \rfloor \left(d \cdot e\right)^{L} \rfloor \epsilon$$

$$(iii) \qquad \left(a \cdot b \cdot c\right)^{L} \rfloor \left(f \cdot g \mid \left(d \cdot e\right)^{L} \rfloor \epsilon\right)$$

Structural Congruence

The **Structural Congruence** relation $\equiv$ is the least congruence relations on terms satisfying the following rules:

$$S_1 \cdot (S_2 \cdot S_3) \equiv (S_1 \cdot S_2) \cdot S_3 \qquad S \cdot \epsilon \equiv_S \epsilon \cdot S \equiv S$$

$$T_1 \mid T_2 \equiv T_2 \mid T_1 \qquad T_1 \mid (T_2 \mid T_3) \equiv (T_1 \mid T_2) \mid T_3$$

$$T \mid \epsilon \equiv T \qquad \left(\epsilon\right)^L \rfloor \epsilon \equiv \epsilon$$

$$\left(S_1 \cdot S_2\right)^L \rfloor T \equiv \left(S_2 \cdot S_1\right)^L \rfloor T$$

Dynamics of the Calculus (1)

$\mathcal{T}_{\mathcal{V}}$ is the set of terms which may contain variables (three kinds):

- term variables $(X, Y, Z, \ldots)$
- sequence variables $(\widetilde{x}, \widetilde{y}, \widetilde{z}, \ldots)$
- element variables $(x, y, z, \ldots)$

$T\sigma$ is the term obtained replacing the variables in T.

A **Rewrite Rule** is a pair (T, T'), denoted $T \mapsto T'$, where:

- $T, T' \in \mathcal{T}_{\mathcal{V}}$ with $T \neq \epsilon$
- variables in T' are a subset of those in T

A rule $T \mapsto T'$ can be applied to all terms $T\sigma$.

Example: $a \cdot x \cdot a \mapsto b \cdot x \cdot b$

- can be applied to $a \cdot c \cdot a$ (producing $b \cdot c \cdot b$)
- cannot be applied to $a \cdot c \cdot c \cdot a$

Dynamics of the Calculus (2)

The semantics of CLS is defined by resorting to the notion of *contexts*.

Contexts $\mathcal{C}$ are given by the following grammar:

$$\mathcal{C} ::= \Box \quad | \quad \mathcal{C} \mid T \quad | \quad T \mid \mathcal{C} \quad | \quad (S)^L \rfloor \mathcal{C}$$

where $T \in \mathcal{T}$ and $S \in \mathcal{S}$. Context $\Box$ is called the *empty context*.

Given a finite set of rewrite rules $\mathcal{R}$, the *reduction semantics* of CLS is the least relation closed with respect to $\equiv$ and satisfying the following inference rule:

$$\frac{T \mapsto T' \in \mathcal{R} \qquad T\sigma \not\equiv \epsilon \qquad \sigma \in \Sigma \qquad C \in \mathcal{C}}{C[T\sigma] \rightarrow C[T'\sigma]}$$

CLS and Biological Interactions (1)

Biomolecular Entity	**CLS Term**
Elementary object (genes, domains, other molecules, etc...)	Alphabet symbol
DNA strand	Sequence of elements repr. genes
RNA strand	Sequence of elements repr. transcribed genes
Protein	Sequence of elements repr. domains or single alphabet symbol
Molecular population	Parallel composition of molecules
Membrane	Looping sequence

Table: Guidelines for the abstraction of biomolecular entities into CLS.

CLS and Biological Interactions (2)

Biomolecular Event	**Examples of CLS Rewrite Rule**
State change	$a \longrightarrow b$ $\widetilde{x} \cdot a \cdot \widetilde{y} \longrightarrow \widetilde{x} \cdot b \cdot \widetilde{y}$ $(a \cdot \widetilde{x})^L \rfloor X \longrightarrow (b \cdot \widetilde{x})^L \rfloor X$
Complexation	$a \mid b \longrightarrow c$ $\widetilde{x} \cdot a \cdot \widetilde{y} \mid b \longrightarrow \widetilde{x} \cdot c \cdot \widetilde{y}$
Decomplexation	$c \longrightarrow a \mid b$ $\widetilde{x} \cdot c \cdot \widetilde{y} \longrightarrow \widetilde{x} \cdot a \cdot \widetilde{y} \mid b$
Catalysis	$c \mid T_1 \longrightarrow c \mid T_2$ where $T_1 \longrightarrow T_2$ is the catalyzed event
Complexation on membrane	$(a \cdot \widetilde{x} \cdot b \cdot \widetilde{y})^L \rfloor X \longrightarrow (c \cdot \widetilde{x} \cdot \widetilde{y})^L \rfloor X$ $a \mid (b \cdot \widetilde{x})^L \rfloor X \longrightarrow (c \cdot \widetilde{x})^L \rfloor X$ $(b \cdot \widetilde{x})^L \rfloor (a \mid X) \longrightarrow (c \cdot \widetilde{x})^L \rfloor X$
Decomplexation on membrane	$(c \cdot \widetilde{x})^L \rfloor X \longrightarrow (a \cdot b \cdot \widetilde{x})^L \rfloor X$ $(c \cdot \widetilde{x})^L \rfloor X \longrightarrow a \mid (b \cdot \widetilde{x})^L \rfloor X$ $(c \cdot \widetilde{x})^L \rfloor X \longrightarrow (b \cdot \widetilde{x})^L \rfloor (a \mid X)$
Catalysis on membrane	$(c \cdot \widetilde{x} \cdot S_1 \cdot \widetilde{y})^L \rfloor X \longrightarrow (c \cdot \widetilde{x} \cdot S_2 \cdot \widetilde{y})^L \rfloor X$ where $S_1 \longrightarrow S_2$ is the catalyzed event

Table: Guidelines for the abstraction of biomolecular events into CLS.

CLS and Biological Interactions (3)

Biomolecular Event	Examples of CLS Rewrite Rule
Membrane crossing	$a \mid (\widetilde{x})^L \rfloor X \longrightarrow (\widetilde{x})^L \rfloor (a \mid X)$ $(\widetilde{x})^L \rfloor (a \mid X) \longrightarrow a \mid (\widetilde{x})^L \rfloor X$ $\widetilde{x} \cdot a \cdot \widetilde{y} \mid (\widetilde{z})^L \rfloor X \longrightarrow (\widetilde{z})^L \rfloor (\widetilde{x} \cdot a \cdot \widetilde{y} \mid X)$ $(\widetilde{z})^L \rfloor (\widetilde{x} \cdot a \cdot \widetilde{y} \mid X) \longrightarrow \widetilde{x} \cdot a \cdot \widetilde{y} \mid (\widetilde{z})^L \rfloor X$
Catalyzed membrane crossing	$a \mid (b \cdot \widetilde{x})^L \rfloor X \longrightarrow (b \cdot \widetilde{x})^L \rfloor (a \mid X)$ $(b \cdot \widetilde{x})^L \rfloor (a \mid X) \longrightarrow a \mid (b \cdot \widetilde{x})^L \rfloor X$ $\widetilde{x} \cdot a \cdot \widetilde{y} \mid (b \cdot \widetilde{z})^L \rfloor X \longrightarrow (b \cdot \widetilde{z})^L \rfloor (\widetilde{x} \cdot a \cdot \widetilde{y} \mid X)$ $(b \cdot \widetilde{z})^L \rfloor (\widetilde{x} \cdot a \cdot \widetilde{y} \mid X) \longrightarrow \widetilde{x} \cdot a \cdot \widetilde{y} \mid (b \cdot \widetilde{z})^L \rfloor X$
Membrane joining	$(\widetilde{x})^L \rfloor (a \mid X) \longrightarrow (a \cdot \widetilde{x})^L \rfloor X$ $(\widetilde{x})^L \rfloor (\widetilde{y} \cdot a \cdot \widetilde{z} \mid X) \longrightarrow (\widetilde{y} \cdot a \cdot \widetilde{z} \cdot \widetilde{x})^L \rfloor X$
Catalyzed membrane joining	$(b \cdot \widetilde{x})^L \rfloor (a \mid X) \longrightarrow (a \cdot b \cdot \widetilde{x})^L \rfloor X$ $(b \cdot \widetilde{x})^L \rfloor (\widetilde{y} \cdot a \cdot \widetilde{z} \mid X) \longrightarrow (\widetilde{y} \cdot a \cdot \widetilde{z} \cdot b \cdot \widetilde{x})^L \rfloor X$
Membrane fusion	$(\widetilde{x})^L \rfloor (X) \mid (\widetilde{y})^L \rfloor (Y) \longrightarrow (\widetilde{x} \cdot \widetilde{y})^L \rfloor (X \mid Y)$
Catalyzed membrane fusion	$(a \cdot \widetilde{x})^L \rfloor (X) \mid (b \cdot \widetilde{y})^L \rfloor (Y) \longrightarrow (a \cdot \widetilde{x} \cdot b \cdot \widetilde{y})^L \rfloor (X \mid Y)$
Membrane division	$(\widetilde{x} \cdot \widetilde{y})^L \rfloor (X \mid Y) \longrightarrow (\widetilde{x})^L \rfloor (X) \mid (\widetilde{y})^L \rfloor (Y)$
Catalyzed membrane division	$(a \cdot \widetilde{x} \cdot b \cdot \widetilde{y})^L \rfloor (X \mid Y) \longrightarrow (a \cdot \widetilde{x})^L \rfloor (X) \mid (b \cdot \widetilde{y})^L \rfloor (Y)$

Table: Guidelines for the abstraction of biomolecular events into CLS.

Bisimulations

Bisimilarity is widely accepted as the finest extensional behavioural equivalence one may impose on systems.

- Two systems are bisimilar if they can perform step by step the same interactions with the environment.
- Properties of a system can be verified by assessing the bisimilarity with a system known to enjoy them.
- A system's state space can be reduced by lumping bisimilar states.

Bisimilarities need a semantics based on labeled transition relations capturing the potential interactions with the environment.

- In process calculi, transitions are usually labeled with actions.
- In CLS labels are contexts in which rules can be applied[1].

[1] P. Sewell: **From Rewrite Rules to Bisimulation Congruences**. Theoretical Computer Science 274, pp. 183-230, 2002.

Labeled Semantics (1)

Contexts $\mathcal{C}$ are given by the following grammar:

$$\mathcal{C} ::= \Box \quad | \quad \mathcal{C} \mid T \quad | \quad T \mid \mathcal{C} \quad | \quad (S)^L \rfloor \mathcal{C}$$

where $T \in \mathcal{T}$ and $S \in \mathcal{S}$. Context $\Box$ is called the *empty context*.

Parallel Contexts $\mathcal{C}_P$ are given by the following grammar:

$$\mathcal{C}_P ::= \Box \quad | \quad \mathcal{C}_P \mid T \quad | \quad T \mid \mathcal{C}_P.$$

where $T \in \mathcal{T}$.

$C[T]$ is context application and $C[C']$ is context composition.

Labeled Semantics (2)

Given a set of rewrite rules $\mathcal{R} \subseteq \Re$, the **labeled semantics** of CLS is the labeled transition system given by the following inference rules:

$$(\text{rule\_appl}) \quad \frac{T \mapsto T' \in \mathcal{R} \quad C[T''] \equiv T\sigma \quad T'' \not\equiv \epsilon \quad \sigma \in \Sigma \quad C \in \mathcal{C}}{T'' \xrightarrow{C} T'\sigma}$$

$$(\text{cont}) \quad \frac{T \xrightarrow{\Box} T'}{(S)^L \rfloor T \xrightarrow{\Box} (S)^L \rfloor T'} \qquad (\text{par}) \quad \frac{T \xrightarrow{C} T' \quad C \in \mathcal{C}_P \quad C[\epsilon] \not\bowtie T''}{T \mid T'' \xrightarrow{C} T' \mid T''}$$

Rule (rule_appl) describes the (potential) application of a rule.

- $T'' \not\equiv \epsilon$ in the premise implies that C cannot provide completely the left hand side of the rewrite rule.
- Example: let $R = a \mid b \mapsto c$, we have $a \xrightarrow{\Box \mid b} c$, but $\epsilon \not\xrightarrow{a|b}$.

Labeled Semantics (3)

Given a set of rewrite rules $\mathcal{R} \subseteq \Re$, the **labeled semantics** of CLS is the labeled transition system given by the following inference rules:

$$(\text{rule\_appl}) \quad \frac{T \mapsto T' \in \mathcal{R} \quad C[T''] \equiv T\sigma \quad T'' \not\equiv \epsilon \quad \sigma \in \Sigma \quad C \in \mathcal{C}}{T'' \xrightarrow{C} T'\sigma}$$

$$(\text{cont}) \quad \frac{T \xrightarrow{\Box} T'}{(S)^L \rfloor T \xrightarrow{\Box} (S)^L \rfloor T'} \qquad (\text{par}) \quad \frac{T \xrightarrow{C} T' \quad C \in \mathcal{C}_P \quad C[\epsilon] \not\sqcap T''}{T \mid T'' \xrightarrow{C} T' \mid T''}$$

Rule (cont) propagates $\Box$–labelled transitions from the inside to the outside of a looping sequence.

- Transition labeled with a non–empty context cannot be propagated.
- Example: let $R = a \mid b \mapsto c$, we have $a \xrightarrow{\Box \mid b} c$, but $(d)^L \rfloor a \not\xrightarrow{\Box \mid b}$.

Labeled Semantics (4)

Given a set of rewrite rules $\mathcal{R} \subseteq \Re$, the **labeled semantics** of CLS is the labeled transition system given by the following inference rules:

$$(\text{rule\_appl}) \quad \frac{T \mapsto T' \in \mathcal{R} \quad C[T''] \equiv T\sigma \quad T'' \not\equiv \epsilon \quad \sigma \in \Sigma \quad C \in \mathcal{C}}{T'' \xrightarrow{C} T'\sigma}$$

$$(\text{cont}) \quad \frac{T \xrightarrow{\Box} T'}{(S)^L \rfloor T \xrightarrow{\Box} (S)^L \rfloor T'} \qquad (\text{par}) \quad \frac{T \xrightarrow{C} T' \quad C \in \mathcal{C}_P \quad C[\epsilon] \not\sqcap T''}{T \mid T'' \xrightarrow{C} T' \mid T''}$$

Rule (par) propagates transitions labelled with parallel contexts in parallel components.

- $C[\epsilon] \not\sqcap T''$ ensures C is the least necessary.
- Example: let $R = a \mid b \mapsto c$, we have $b \xrightarrow{a \mid \Box} c$, but $b \mid a \not\xrightarrow{a \mid \Box} c \mid a$.

Bisimulations in CLS (1)

A binary relation R on terms is a **strong bisimulation** if, given T_1, T_2 such that $T_1 R T_2$, the two following conditions hold:

- $T_1 \xrightarrow{C} T_1' \implies \exists T_2'$ s.t. $T_2 \xrightarrow{C} T_2'$and $T_1' R T_2'$
- $T_2 \xrightarrow{C} T_2' \implies \exists T_1'$ s.t. $T_1 \xrightarrow{C} T_1'$ and $T_2' R T_1'$.

The *strong bisimilarity* $\sim$ is the largest of such relations.

A binary relation R on terms is a **weak bisimulation** if, given T_1, T_2 such that $T_1 R T_2$, the two following conditions hold:

- $T_1 \xrightarrow{C} T_1' \implies \exists T_2'$ s.t. $T_2 \stackrel{C}{\Longrightarrow} T_2'$and $T_1' R T_2'$
- $T_2 \xrightarrow{C} T_2' \implies \exists T_1'$ s.t. $T_1 \stackrel{C}{\Longrightarrow} T_1'$ and $T_2' R T_1'$.

The *weak bisimilarity* $\approx$ is the largest of such relations.

Theorem: Strong and weak bisimilarities are congruences.

Bisimulations in CLS (2)

Consider the following set of rewrite rules:

$$\mathcal{R} = \{ \quad a \mid b \mapsto c \ , \ d \mid b \mapsto e \ , \ e \mapsto e \ , \ c \mapsto e \ , \ f \mapsto a \ \}$$

We have that $a \sim d$, because

$$a \xrightarrow{\square|b} c \xrightarrow{\square} e \xrightarrow{\square} e \xrightarrow{\square} \ldots$$

$$d \xrightarrow{\square|b} e \xrightarrow{\square} e \xrightarrow{\square} \ldots$$

and $f \approx d$, because

$$f \xrightarrow{\square} a \xrightarrow{\square|b} c \xrightarrow{\square} e \xrightarrow{\square} e \xrightarrow{\square} \ldots$$

On the other hand, $f \not\sim e$ and $f \not\approx e$.

$$e \xrightarrow{\square} e \xrightarrow{\square} e \xrightarrow{\square} \ldots$$

A Quantitative Analysis

Chemical reactions are described by the *law of mass action*

- the speed of a reaction is proportional to the concentrations of the individual reactants involved
- differential equations

The simulation algorithm we propose

- introduces reaction speeds in the rewrite rules for CLS terms;
- assumes that at each step at most one reaction may occur (randomly chosen).

The Law of Mass Action

Usual notation for chemical reactions:

$$\ell_1 S_1 + \ldots + \ell_\rho S_\rho \xrightarrow{k} \ell'_1 P_1 + \ldots + \ell'_\gamma P_\gamma$$

where:

- S_i, P_i are molecules
- ℓ_i, ℓ'_i are stoichiometric coefficients
- k is the kinetic constant

For the *law of mass action*, the rate of production is:

$$\frac{dP_i}{dt} = k\ell'_i[S_1]^{\ell_1} \cdots [S_\rho]^{\ell_\rho}$$

Stochastic Rewrite Rules

A **Stochastic Rewrite Rule** is a triple (T, k, T') where:

- $T, T' \in \mathcal{T}_{\mathcal{V}}$ with $T \neq \epsilon$ are the source and the target term, respectively;
- variables in T' are a subset of those in T;
- k is a rate modelling the parameter of an exponential probability distribution representing the speed of the rule;

Example: $a|b \overset{k}{\mapsto} a \cdot b$

- reactants $a|b$ are linked together in the sequence $a \cdot b$
- k models the kinetic constant of the reaction described by the rule

Dynamics of the Stochastic Calculus

- Enumeration of the possible occurrences of the LHS of a rule (the higher this value, the faster the reaction).
- At each step one of the rules in $\mathcal{R}$ is probabilistically chosen and applied somewhere in T obtaining a new term T'.
- The duration of the reaction is chosen probabilistically according to the exponential distribution.
- Many different evolutions of a term are possible, each one with different probabilities.

Example: $\mathcal{R} = \{ \quad a|b \overset{k_1}{\mapsto} c \quad , \quad \left(\widetilde{x}\right)^L \rfloor (X|c) \overset{k_2}{\mapsto} \left(\widetilde{x} \cdot c\right)^L \rfloor X \quad \}$

$$T = \left(m\right)^L \rfloor (a|a|a|b|b) \quad \overset{6 \cdot k_1}{\mapsto} \quad \left(m\right)^L \rfloor (a|a|c|b) \quad \overset{k_2}{\mapsto}$$

$$\overset{k_2}{\mapsto} \quad \left(m \cdot c\right)^L \rfloor (a|a|b) \quad \rightarrow \ldots$$

Stochastic Semantics for CLS

Given a finite set of rewrite rules $\mathcal{R}$, the *stochastic reduction semantics* of Stochastic CLS is the least relation satisfying the following inference rule:

$$\frac{R_i : T_i \overset{k}{\mapsto} T_i' \qquad T \equiv C[T_i]}{T \xrightarrow{R_i, k \cdot occ(R_i, T, C[T_i'])} C[T_i']}$$

Our stochastic reduction semantics is essentially a *Continuous–time Markov Chain* (CTMC).

Proposition: Let $\mathcal{R}$ be a set of stochastic rewrite ruled, and let $\mathcal{R}'$ be the set of rewrite rules of the standard CLS obtained by removing all rate functions. It holds: $T \rightarrow T' \iff T \xrightarrow{R_i, r} T'$.

An Application to the Modeling of Gene Regulation

Bacteria react to changes in the environment through changes in the enzymes they produce:

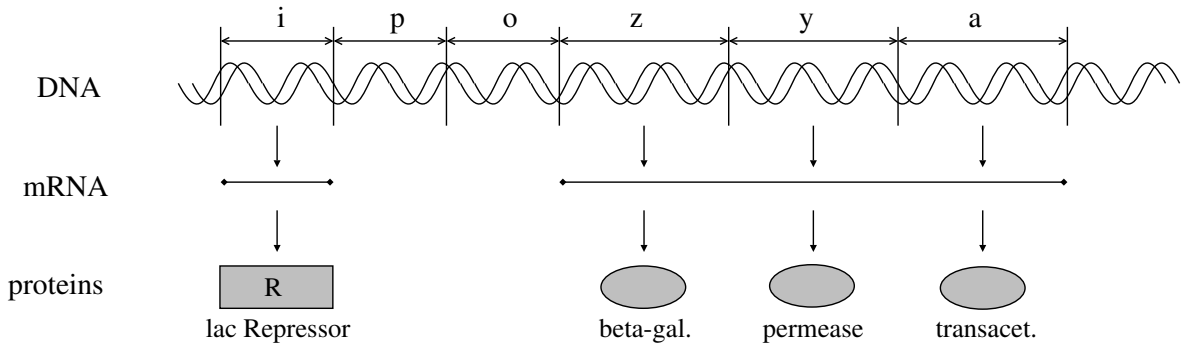
- they do not synthesize degradative enzymes unless the substrates for these enzymes are present in the environment.

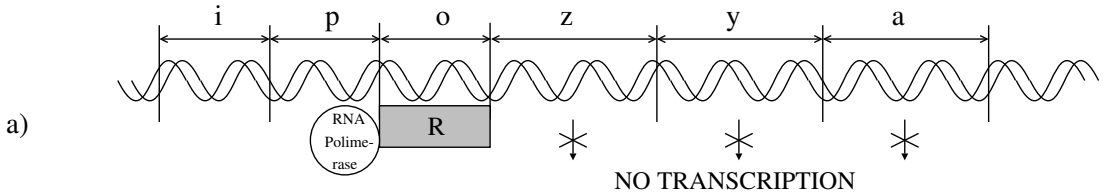
E. coli does not sinthesize the enzymes that degrade lactose unless lactose is in the environment. This phenomenon is called **gene regulation** since it is obtained by controlling the transcription of some genes into the corresponding enzymes.

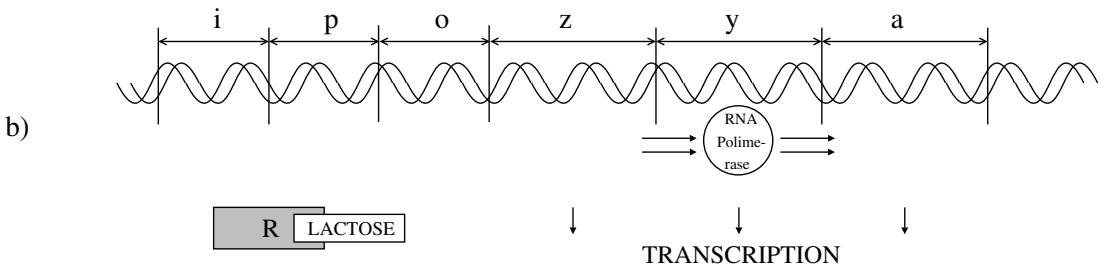
We use the Stochastic CLS:

- to describe the regulation process of the **lactose operon** in E. coli;
- to use our stochastic simulator for analyzing the gene regulation process in different situations.

The Lactose Operon in E.coli







The Stochastic CLS Model (1)

Notation:

- $n \times T := T \mid \ldots \mid T$;
- $(m)^L :=$ membrane of the bacterium;
- $lacI{-}A := lacI \cdot lacP \cdot lacO \cdot lacZ \cdot lacY \cdot lacA$.

The initial state of the bacterium when no lactose is present in the environment is modeled by the following term :

$$Ecoli \ ::= \ (m)^L \rfloor (lacI{-}A \mid 30 \times polym \mid 100 \times repr) \qquad (1)$$

The presence of lactose is modeled by composing *Ecoli* in parallel with a number of *LACT* elements:

$$EcoliLact \ ::= \ Ecoli \mid 100 \times LACT \qquad (2)$$

The Stochastic CLS Model (2)

Transcription of DNA, binding of lac Repressor to gene o, and interaction between lactose and lac Repressor:

$$lacI \cdot \widetilde{x} \xrightarrow{0.02} lacI \cdot \widetilde{x} \mid Irna \quad \text{(S1)}$$

$$Irna \xrightarrow{0.1} Irna \mid repr \quad \text{(S2)}$$

$$polym \mid \widetilde{x} \cdot lacP \cdot \widetilde{y} \xrightarrow{0.1} \widetilde{x} \cdot PP \cdot \widetilde{y} \quad \text{(S3)}$$

$$\widetilde{x} \cdot PP \cdot \widetilde{y} \xrightarrow{0.01} polym \mid \widetilde{x} \cdot lacP \cdot \widetilde{y} \quad \text{(S4)}$$

$$\widetilde{x} \cdot PP \cdot lacO \cdot \widetilde{y} \xrightarrow{20.0} polym \mid Rna \mid \widetilde{x} \cdot lacP \cdot lacO \cdot \widetilde{y} \quad \text{(S5)}$$

$$Rna \xrightarrow{0.1} Rna \mid betagal \mid perm \mid transac \quad \text{(S6)}$$

$$repr \mid \widetilde{x} \cdot lacO \cdot \widetilde{y} \xrightarrow{1.0} \widetilde{x} \cdot RO \cdot \widetilde{y} \quad \text{(S7)}$$

$$\widetilde{x} \cdot RO \cdot \widetilde{y} \xrightarrow{0.01} repr \mid \widetilde{x} \cdot lacO \cdot \widetilde{y} \quad \text{(S8)}$$

$$repr \mid LACT \xrightarrow{0.005} RLACT \quad \text{(S9)}$$

$$RLACT \xrightarrow{0.1} repr \mid LACT \quad \text{(S10)}$$

The Stochastic CLS Model (3)

The following schemata describe the behaviour of the three enzymes for lactose degradation:

$$\left(\widetilde{x}\right)^{L} \rfloor \left(perm \mid X\right) \xmapsto{0.1} \left(perm \cdot \widetilde{x}\right)^{L} \rfloor X \tag{S11}$$

$$LACT \mid \left(perm \cdot \widetilde{x}\right)^{L} \rfloor X \xmapsto{0.001} \left(perm \cdot \widetilde{x}\right)^{L} \rfloor \left(LACT \mid X\right) \tag{S12}$$

$$betagal \mid LACT \xmapsto{0.001} betagal \mid GLU \mid GAL \tag{S13}$$

The Stochastic CLS Model (4)

The following schemata describe degradation of all the proteins and pieces of mRNA involved in the process:

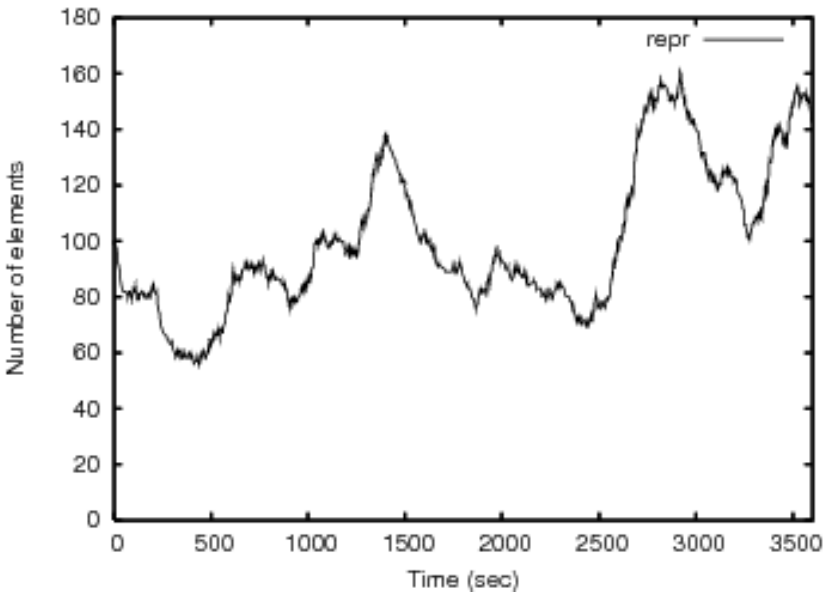
$$perm \overset{0.001}{\mapsto} \epsilon \quad \text{(S14)} \qquad betagal \overset{0.001}{\mapsto} \epsilon \quad \text{(S15)}$$

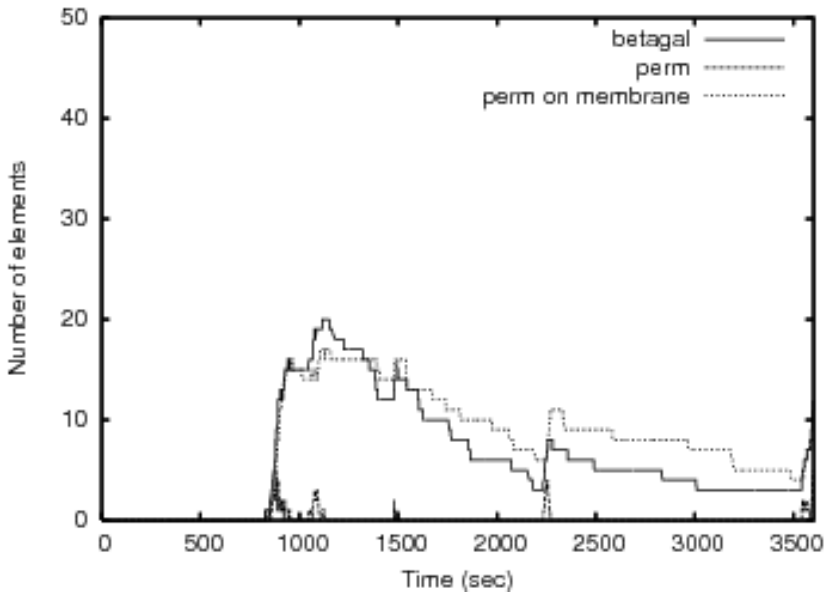
$$transac \overset{0.001}{\mapsto} \epsilon \quad \text{(S16)} \qquad repr \overset{0.002}{\mapsto} \epsilon \quad \text{(S17)}$$

$$Irna \overset{0.01}{\mapsto} \epsilon \quad \text{(S18)} \qquad Rna \overset{0.01}{\mapsto} \epsilon \quad \text{(S19)}$$

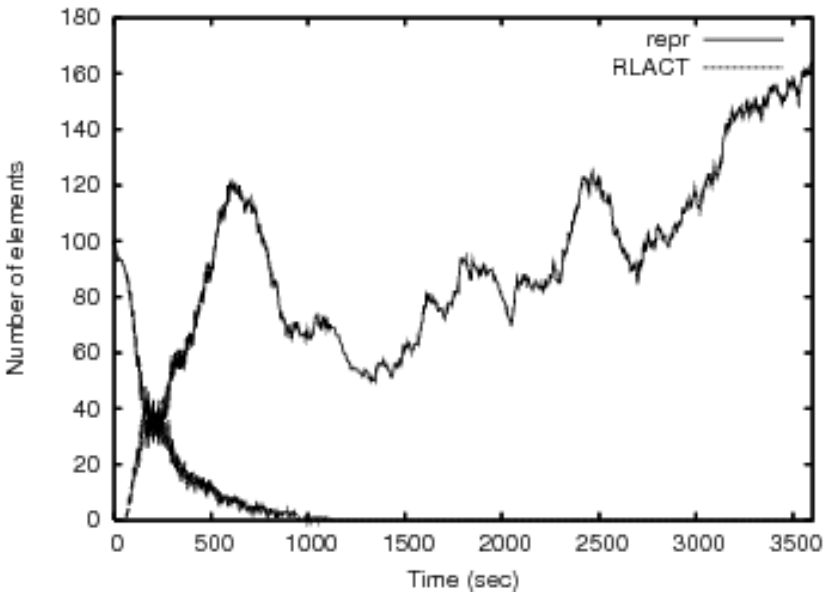
$$RLACT \overset{0.002}{\mapsto} LACT \quad \text{(S20)}$$

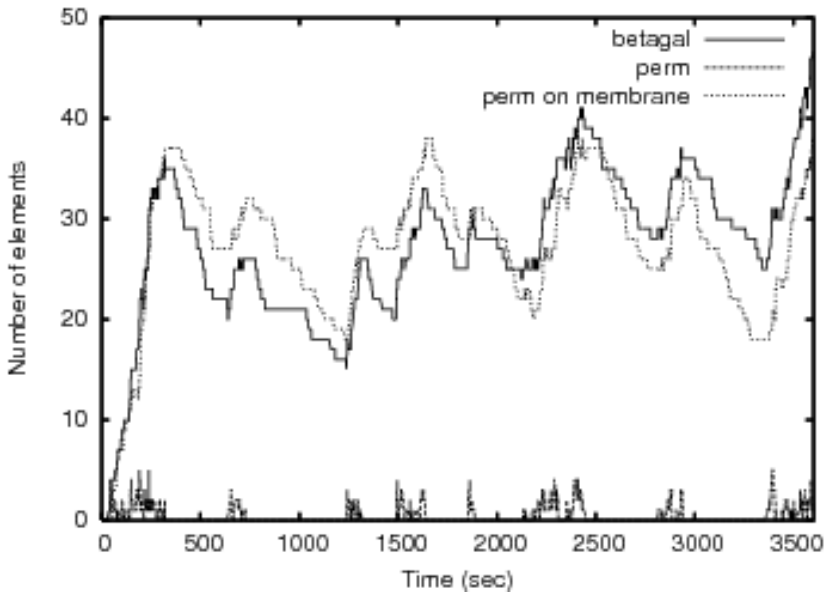
Simulation Results - No Lactose



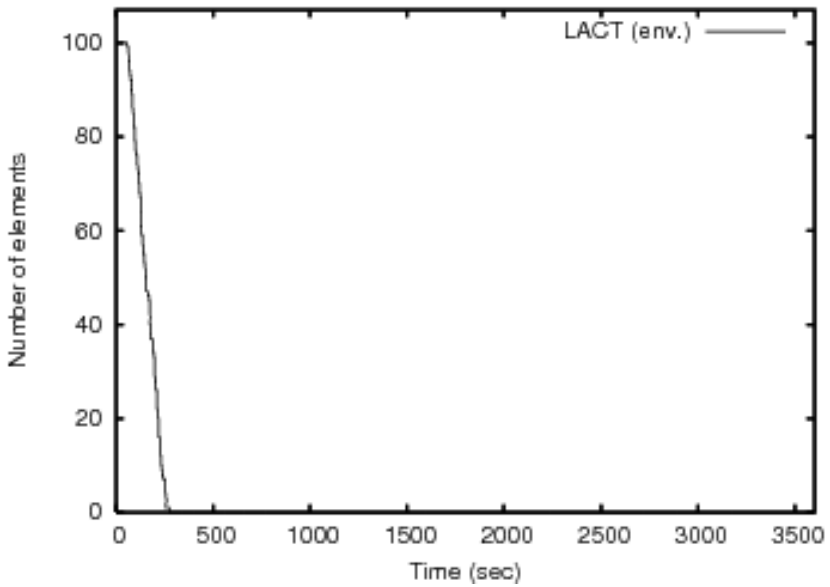


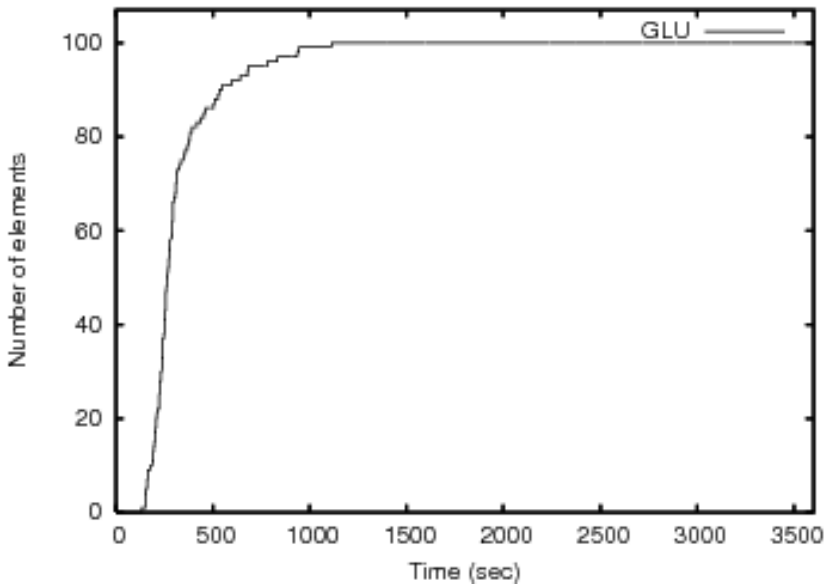
Simulation Results - Lactose





Simulation Results - Lactose





Discussion

Among the formalisms to describe membrane systems we mentioned **Brane Calculi** and **P-Systems**.

CLS can describe situations which cannot be easily captured by these formalisms (which consider membranes as atomic objects).

An example of this is given by the representation of the membrane of E. coli:

- Representing the membrane as a sequence of elements allows the definition of different functionalities depending on type and number of elements constituting the membrane itself;
- The presence and the number of lactose permease on the bacterium membrane regulates the transportation of lactose inside the bacterium.

Summary

The formal modelling of biological systems allows:

1. verification of systems' properties;
2. analysis with simulators;
3. prediction of unknown behaviour.

The Calculus of Looping Sequences can be used to describe biological systems.

The bisimulation relations we have defined can be used:

- to find equivalent reduced models;
- to verify properties.

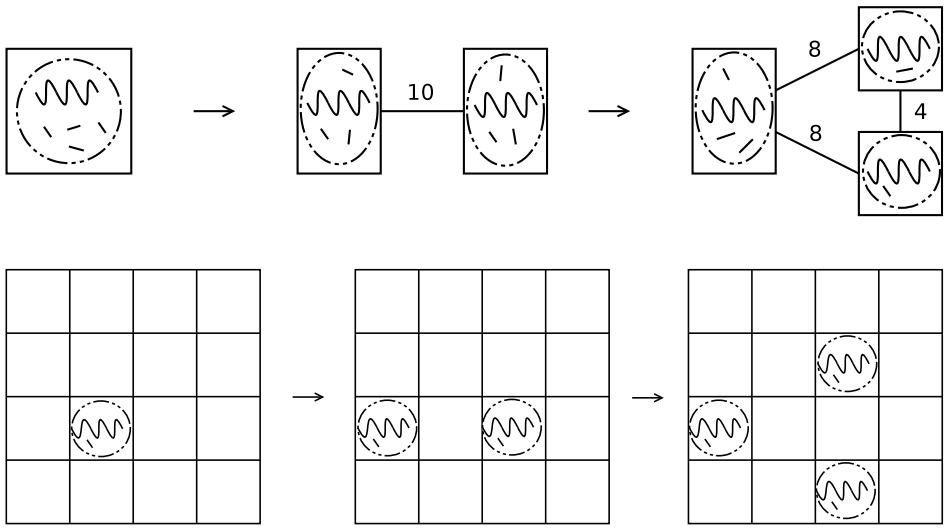
The Stochastic CLS can be used:

- to model quantitative interactions;
- to simulate real experiments.

Modelling Topology: A Possible Extension

Define a "topological" extension of CLS to model cell division and differentiation, tissues, etc. . . :

- the distance between reactants alters the probability of a reaction;
- cells show an inherent topological behaviour (e.g. the neurons differentiation of retinal cells).



References

R. Barbuti, S. Cataudella, A. Maggiolo–Schettini, P. Milazzo and A. Troina. **A probabilistic model for molecular systems**. *Fundamenta Informaticae* 67:13–27, 2005.

R. Barbuti, A. Maggiolo–Schettini, P. Milazzo and A. Troina. **An alternative to Gillespie's algorithm for simulating chemical reactions**. Proc. of CMSB'05, 2005.

R. Barbuti, A. Maggiolo–Schettini, P. Milazzo and A. Troina. **A calculus of looping sequences for modelling microbiological systems**. *Fundamenta Informaticae* 72:21–35, 2006.

R. Barbuti, A. Maggiolo–Schettini, P. Milazzo and A. Troina. **Bisimulation congruences in the calculus of looping sequences**. Proc. of ICTAC'06, Springer LNCS 4281:93–107, 2006.

R. Barbuti, A. Maggiolo–Schettini, P. Milazzo and A. Troina. **The Calculus of Looping Sequences for Modeling Biological Membranes**. Proc. of WMC, invited paper, Springer LNCS 4860, pp. 54-76, 2007.

P. Milazzo. **Qualitative and Quantitative Formal Modeling of Biological Systems**. Ph.D. Thesis, Università di Pisa, 2007.

R. Barbuti, A. Maggiolo–Schettini, P. Milazzo and A. Troina. **Bisimulations in Calculi Modelling Membranes**. *Formal Aspects of Computing*, to appear.

CLS based simulators for biological systems: `http://www.di.unipi.it/~milazzo/biosims/`.